REMARKS

Claims 1, 18, 25, and 41 have been amended to clarify the invention. Claims 1-8 and 10-41 remain pending.

The Examiner rejected claims 1-8 and 10-41 under 35 U.S.C. §103(a) as being unpatentable over Li (U.S. Patent 5,643,125) and further in view of Dreszer (U.S. Patent 6,442,661). The Examiner's rejections are respectfully traversed as follows.

Claim 1 is directed towards a "method for assigning traffic buckets to a cache system." Claim also requires that "a) when a new cache system *starts up in a cache cluster* having a *plurality of cache systems among which a plurality of total buckets are to be allocated,* determining a full bucket allocation for the new cache system; b) *periodically* determining a load of the new cache system; c) when it is determined that the new cache system is underloaded, slowly assigning a portion of the full bucket allocation or a portion of previously shed buckets to the new cache system unless the full bucket allocation has already been assigned to the new cache system; and d) when it is determined that the new cache system is overerloaded, shedding a portion of the buckets previously assigned to the new cache system." Independent claims 18, 25, and 41 recite mechanisms for *periodically* determining a load of a new cache system *that starts up in a cache cluster having a plurality of cache systems among which a plurality of total buckets* are to be allocated and then slowly assigning and shedding buckets to and from the new cache system based on the *periodic* load determination for the new cache system.

The present invention allows efficient assignment of buckets among a plurality of cache systems of a cache cluster system by periodically monitoring the load of a new cache system *when it starts up in a cache cluster having a plurality of cache systems.* Based on this *periodic* monitoring of the new cache system's load, buckets are slowly assigned to this new cache system, which allows the new cache system to not be quickly overwhelmed by too many buckets. Additionally based on this *periodic* monitoring of the new cache system's load, buckets are shed from this new cache system when it is *periodically* determined that the new cache becomes overloaded, which allows the new cache system to not remain overloaded. By applying this technique to each new cache system as it enters the cache cluster, buckets can be *periodically* distributed among the cache systems of the cache cluster while managing the load of individual cache systems so they don't remain over- or underloaded.

The primary reference Li describes a parallel database system for storing tables at multiple nodes. Li also describes techniques for handling a node that is being added to the parallel database system in Col. 5, Line 38 through Col. 6, Line 52.

A data redistribution process for adding a new node is described in Fig. 6 and Col. 6, Lines 18-52. First, a new node is physically added in step 152 of the process. In a next step 154, Li then discloses that "buckets of data to be moved to the new node are determined for each existing node." See Col. 6, Lines 18-26. The buffer size for the logical links of each node is then determined in step 156. See Col. 6, Lines 27-29. Li then describes a table scan process that is performed on each table of each existing node (excluding the new node). This table scan process includes copying records for buckets which are to redistributed to the new node (as determined by step 154 of Fig. 6) into the communication buffer and then marking these copied records to be deleted at the end of the redistribution process. See Fig. 8 and Col. 8, Lines 51-64. These copied records are then redistributed to the new node and deleted from the existing nodes in steps 159 through 170 of Fig. 6. See Col. 6, Lines 36-52.

In other words, when a new node is added, the data is only redistributed once from the existing nodes to the new node. That is, the Li fails to teach or suggest *periodically* monitoring the load of the new node and (i) when it is determined (during this periodic monitoring) that the new node is underloaded, assigning buckets and (ii) when it is determined(during this periodic monitoring) that the new node is overloaded, shedding buckets from such new node, in the manner claimed. In other words, Li fails to teach or suggest periodically determining the load of the new node and then taking action based on this periodic determination, *e.g.,* by assigning buckets when it is periodically determined that the new node is underloaded, in the manner claimed. In other words, Li fails to teach or suggest *periodically* adjusting the new node's load based on a *periodic* monitoring of such load, in the manner claimed.

The secondary reference Dreszer also suffers from the above described deficiencies. In general, Dreszer is directed towards managing a memory cache of a single system and not assigning buckets to a new cache system that is starting up in a cache system having multiple cache systems among which buckets are allocated, in the manner claimed. Even if one were to argue that Dreszer teaches slowing assigning buckets to the cache and shedding buckets from the cache (as argued by the Examiner), this feature cannot be combined with the teachings of Li and result in the present invention. This is because Li does not include a mechanism for periodically monitoring the load of a new cache for the purposes of periodically altering the load of the new cache when such new cache becomes over- or underloaded. That is, the teachings of Li describe only a single bucket determination step for a new node and not a periodic determination of the new node's load, in the manner claimed. Thus, the combination of Li and Dreszer would fail to teach or suggest periodically determining the load of the new node and then taking periodic action based on this determination, *e.g.,* by assigning buckets if the new node is underloaded, in the manner claimed. In other words, Li fails to teach or suggest *periodically*

adjusting the new node's load based on a *periodic* monitoring of such load, in the manner claimed.

For the forgoing reasons, it is respectfully submitted that claims 1, 18, 25, and 41 are patentable over the cited references.

The Examiner's rejections of the dependent claims are also respectfully traversed. However, to expedite prosecution, all of these claims will not be argued separately. Claims 2-9, 11-17, 19-24, and 26-40 each depend directly from independent claims 1, 18, or 25 and, therefore, are respectfully submitted to be patentable over cited art for at least the reasons set forth above with respect to claims 1, 18, and 25. Further, the dependent claims require additional elements that when considered in context of the claimed inventions further patentably distinguish the invention from the cited art.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Mary Ramos Olynick
Reg. 42,963

P.O. Box 778
Berkeley, CA 94704-0778
(510) 843-6200